



Firebird performance counters in details

Dmitry Yemanov
<mailto:dimitr@firebirdsql.org>

Firebird Project
<http://www.firebirdsql.org/>

Thank you



Analysing bottlenecks

Disk

Database

Temporary files

Analysing bottlenecks

Disk

Database

Temporary files

Memory

Static (cache)

Dynamic (pools)

Shared

Analysing bottlenecks

CPU, waits, etc

Real execution time

User/kernel time

Was time spent for work or waiting?

What operations were performed?

Legacy performance counters

Page level

Fetches, reads, marks, writes

Shared in SS, per connection in CS

Available via API

Legacy performance counters

Page level

Fetches, reads, marks, writes

Shared in SS, per connection in CS

Available via API

Record level

Selected (seq/idx), inserted, updated, deleted etc

Per connection and per table

Available via API

Improvements in Firebird 2.x

Multi-level aggregated counters

Database (SS only)

Connection

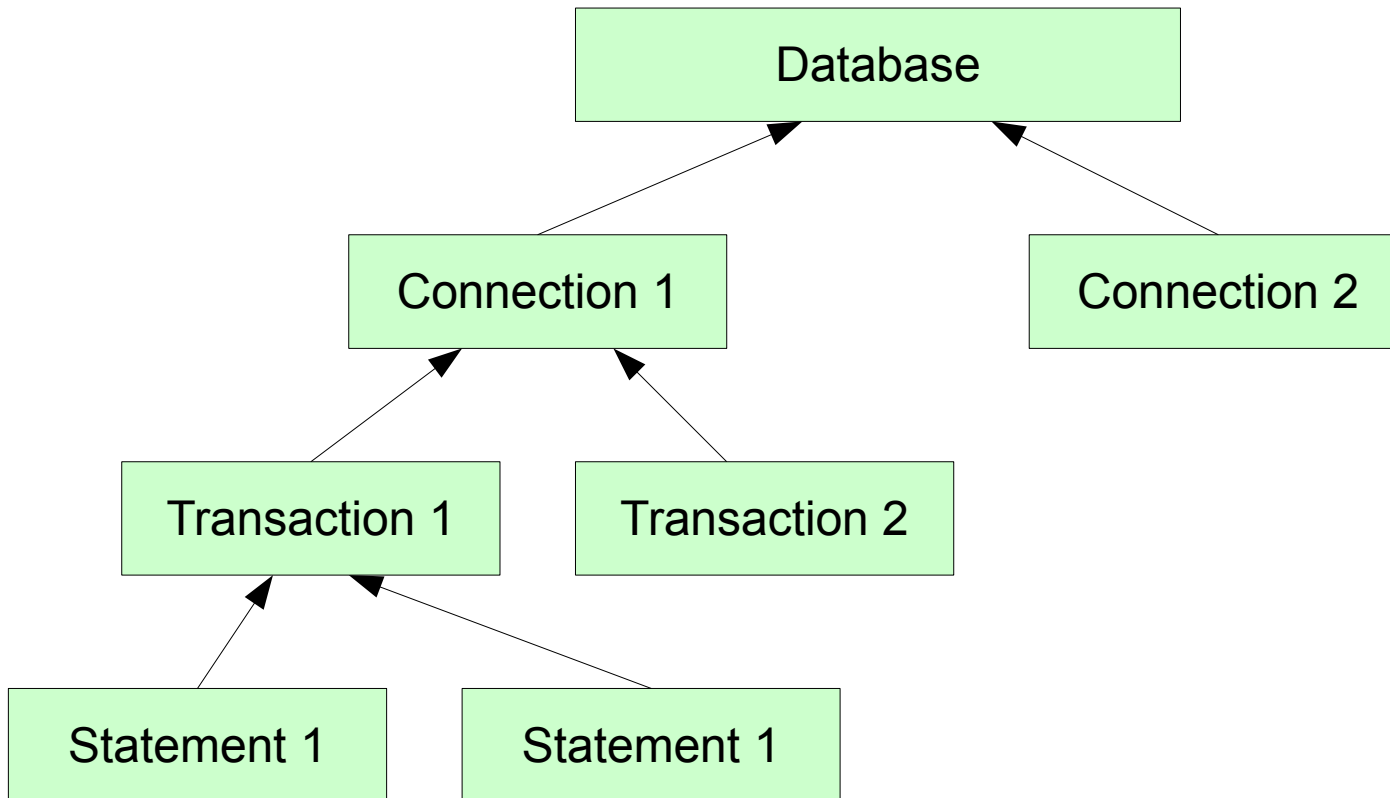
Transaction

Statement

Nested PSQL call (procedure / trigger)

Improvements in Firebird 2.x

Multi-level aggregated counters



Improvements in Firebird 2.x

New interfaces

Monitoring tables

Trace/audit

Page level counters

Page reads

From disk to the page cache (physical reads)

Usually means a cache miss

Page level counters

Page reads

From disk to the page cache (physical reads)

Usually means a cache miss

Page fetches

Access page in cache (logical reads)

Includes both cache hits and cache misses

Corresponds to a shared page lock / latch

Page level counters

Page reads

From disk to the page cache (physical reads)

Usually means a cache miss

Page fetches

Access page in cache (logical reads)

Includes both cache hits and cache misses

Corresponds to a shared page lock / latch

Cache hit ratio = $1 - \text{reads} / \text{fetches}$???

Page level counters

Page writes

From the page cache to disk (physical writes)

At transaction commit/rollback

Cache is full of dirty pages

Asynchronous notification is received (CS/SC)

Immediately after modification

Page level counters

Page writes

From the page cache to disk (physical writes)

At transaction commit/rollback

Cache is full of dirty pages

Asynchronous notification is received (CS/SC)

Immediately after modification

Page marks

Access page in cache (logical writes)

Corresponds to an exclusive page lock / latch

Record level counters

Sequential record reads

Records retrieved through a full table scan

Includes sweep

Indexed record reads

Records retrieved positionally

Includes bitmap index scans, index navigational walks, DBKEY based retrievals

Record level counters

Record inserts, updates, deletes

Pretty obvious, huh?

Record level counters

Record inserts, updates, deletes

Pretty obvious, huh?

Record backouts

Latest (uncommitted) version removed

Happens after savepoint rollback, explicit or implicit

May happen after transaction rollback

Record level counters

Record purges

Old versions removed while keeping the primary version in place

Outdated versions found while chasing

Record level counters

Record purges

Old versions removed while keeping the primary version in place

Outdated versions found while chasing

Record expunges

Whole version chains removed, along with the primary version

Record is deleted and nobody is interested

New record level counters in v3.x

Record repeated reads

Record is retrieved multiple times

BEFORE triggers

Sort-based updates/deletes

New record level counters in v3.x

Record repeated reads

Record is retrieved multiple times

BEFORE triggers

Sort-based updates/deletes

Record locks

Record is selected using WITH LOCK clause

New record level counters in v3.x

Record waits

Attempts to update/delete/lock record
owned by a concurrent active transaction

Transaction is in the WAIT mode

New record level counters in v3.x

Record waits

Attempts to update/delete/lock record owned by a concurrent active transaction

Transaction is in the WAIT mode

Record conflicts

Unsuccessful attempts to update/delete/lock record owned by a concurrent active transaction

UPDATE CONFLICT is reported

New record level counters in v3.x

Backversion reads

Versions chased while finding a visible one

Means old snapshots

New record level counters in v3.x

Backversion reads

Versions chased while finding a visible one

Means old snapshots

Fragment reads

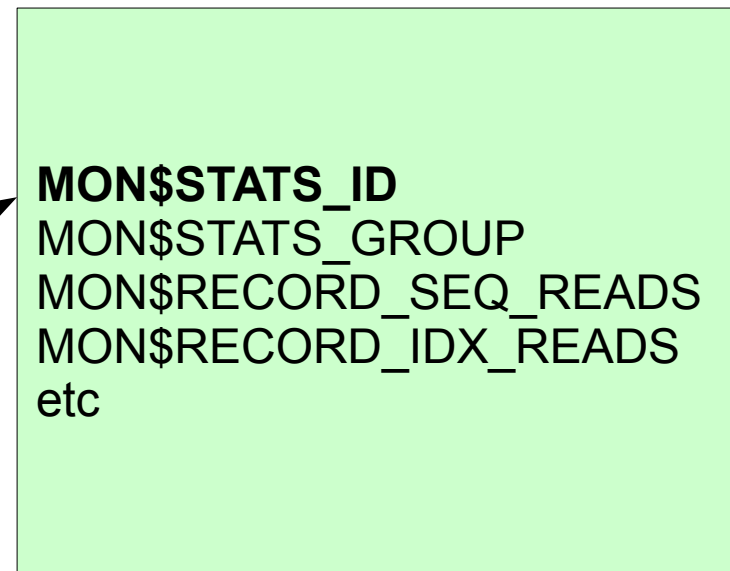
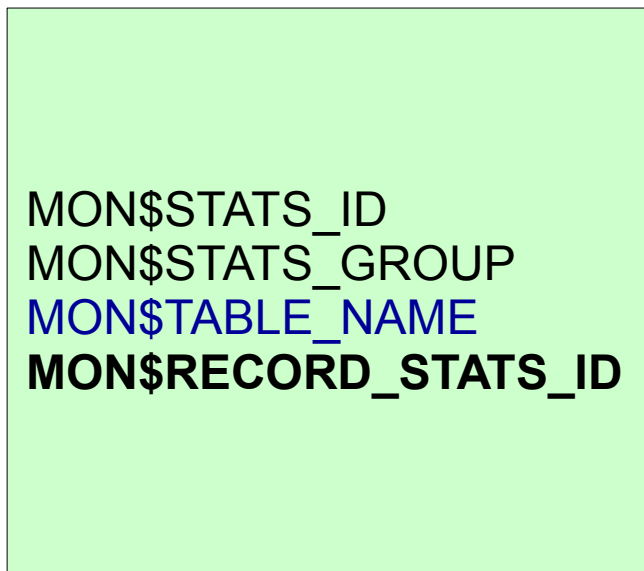
Fragmented records

Means extra page fetches/reads

New record level counters in v3.x

MON\$TABLE_STATS

MON\$RECORD_STATS



TODO: page level counters

Page writes (expanded)

Regular writes (immediate / commit / rollback)

Overflow writes

Asynchronous writes

TODO: page level counters

Page writes (expanded)

Regular writes (immediate / commit / rollback)

Overflow writes

Asynchronous writes

Page waits

How many times page requests waited

Shows contention inside the page cache

TODO: index counters

Possible metrics

Index scans

Node inserts / deletes

Bucket splits / merges

Keys scanned / compared while searching

Reported per index

MON\$INDEX_STATS

TODO: temporary space counters

Operation metrics

Reads / writes resolved through the cache

Reads / writes redirected to temp files

I/O amount metrics

Bytes read / written through the cache

Bytes read / written from/to temp files

TODO: time statistics

Elapsed time

- Total inside the engine

- Spent in the user space

- Spent in the system / kernel space



TODO: time statistics

Elapsed time

Total inside the engine

Spent in the user space

Spent in the system / kernel space

Wait time

Database (and maybe temp files) I/O

Page cache waits

Transaction waits

TODO: integration

Collecting the statistics

StatsD, CollectD

Reporting / analyzing

Graphite / Graphene

Other tools

Nagios, Cacti, Zabbix



Questions?

<mailto:dimitr@firebirdsql.org>